VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Návrh jednotného autentizačního systému založeného na standardu SAML

Design of Single Sign On Based on the SAML Standard

2020 Bc. Jan Javorek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Jan Javorek**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Návrh jednotného autentizačního systému založeného na standardu SAML
Design of Single Sign On Based on the SAML Standard

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem diplomové práce je navrhnout a otestovat funkčnost jednotného autentizačního systému na bázi standardu SAML.

Řešení práce spočívá ve splnění následujících bodů:
1. Popis standardu SAML.
2. Popis různých autentizačních mechanismů, srovnání.
3. Návrh jednotného autentizačního systému SSO.
4. Ověření funkčnosti minimálně na dvou Service Provider serverech.
5. Testování navrženého řešení z hlediska využívání systémových prostředků.

Seznam doporučené odborné literatury:

[1] GERARDUS, Blokdyk *Single sign-on Complete Self-Assessment Guide.*CreateSpace Independent Publishing Platform 2017, ISBN-13: 978-1974021765
[2] GERARDUS, Blokdyk *SAML 2.0 A Complete Guide.* 5STARCooks 2019, ISBN-13: 978-0655535508

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 30. 4. 2020 .....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Diplomová práce se zaměřuje na návrh, vytvoření a ověření funkčnosti autentizačního systému jednotného přihlašování s využitím standardu SAML. Obsahuje popis použitých nástrojů, rozbor standardu SAML, podrobnou konfiguraci a ukázku funkčnosti autentizačního systému. Systém je plně funkční vytvořený v rámci virtualizace a připravený na praktické nasazení.

Klíčová slova: SSO, autentizace, SAML, systém jednotného přihlašování, autentizační systém

Abstract

The diploma thesis focuses on the design, creation and verification of the functionality of the single sign-on authentication system using the SAML standard. It contains a description of the tools used, an analysis of the SAML standard, a detailed configuration and a demonstration of the authentication system functionality. The system is fully functional created within virtualization and ready for practical deployment.

Keywords: Single Sing-On, authentication, SAML, SSO, authentication system

Obsah

Seznam použitých zkratek a symbolů

SSO	–	Single sign-on - systém jednotného přihlášení
OAuth	–	Open Authorization
SAML	–	Security Assertion Markup Language
SP	–	Service provider - poskytovatel služeb
IdP	–	Identity provider - poskytovatel identity
XML	–	eXtensible Markup Language
CPU	–	Central processing unit
LDAP	–	Lightweight Directory Access Protocol
KDC	–	Key distribution center - Centrum distribuce klíčů
HTTP	–	Hypertext Transfer Protocol
HTTPS	–	Hypertext Transfer Protocol Secure
SOAP	–	Simple Object Access Protocol
SSH	–	Secure shell
PHP	–	Hypertext Preprocessor
SSL	–	Secure Sockets Layer
MVC	–	Model-View-Controller
LXC	–	Linux Containers

Seznam obrázků

Seznam tabulek

1 Úvod

Autentizace uživatelů je v dnešní době často řešeným tématem. Možností jak autentizaci udělat bezpečnější a zároveň ji usnadnit uživatelům je několik. Jedním z nich se zabývá tato práce. Jde o jednotný autentizační systém, ten je velmi rozšířený právě díky bezpečnosti a jednoduššímu přistupu pro uživatele. Uživatel se potom dokáže autentizovat vůči různým internetovým službám bezpečně jedním účtem, a volně mezi nimi přecházet. K řešení této problematiky se nabízí mnoho standardů, práce se však zaměřuje na otevřený standard SAML.

Cílem této práce je vytvořit návrh a ověřit funkčnost jednotného autentizační systému založeném právě na standardu SAML. Ten je zapotřebí popsat a srovnat s jinými systémy, které se tímto zabývají. V neposlední řadě je potřeba systém navrhnout a ověřit jeho funkčnost včetně testování na využívání systémových prostředků.

V prvních kapitovách se práce věnuje výhodám a riziků systému jednotného přihlašování, mezi velkou výhodu a zároveň riziko patří například, využití jednoho hesla pro přihlašování uživatele k více internetovým službám. Potenciální útočník se pak ziskem jednoho hesla dostane ke všem službám v rámci tohoto systému. Na druhou stranu pro uživatele je využívání jednoho hesla mnohem přívětivější.

Další část je věnovaná různým metodám implementace tohoto SSO systému. Jsou to metody založené na federaci identity, protokolu Kerberos, biometrii nebo třeba Cookie souborech. V dnešní době jsou však nejvíce využívané první dvě zmíněné metody, to znamená metoda federace identity a systém Kerberos.

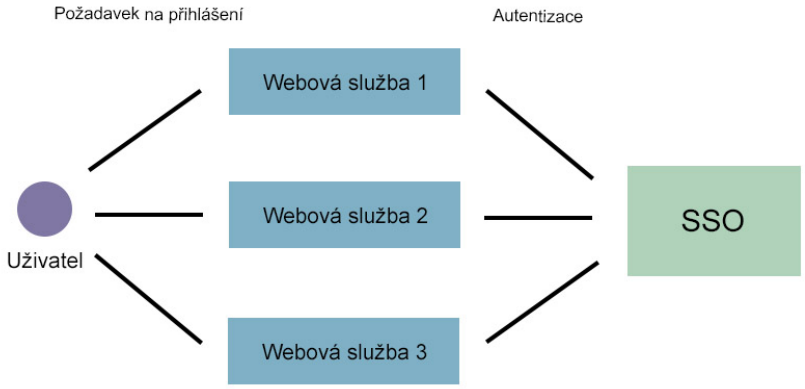
Podstatnou částí práce je popis standardu SAML. Je zde popsaná jeho architektura, výměna zpráv mezi poskytovateli a ostatní podstatné informace. Mezi hlavní důvody proč je vhodné použít právě standard SAML je to, že SAML v první řadě sjednocuje standard pro výměnu autentizačních informací pro SSO systém. SAML poskytuje stejnou identitu pro uživatele napříč různými službami a navíc SAML díky své modularitě nabízí využití v různých webových službách, které nejsou založené na tomto standardu.

Dále jsou uvedeny použité nástroje k vyhotovení SSO systému, jeho návrh a popis konfigurace. Systém je implementovaný s využitím virtualizace, konkrétně linuxových kontejnerů, kterých je pět a každý z nich má jinou funkci. Jeden slouží jako autentizační server, další dva znich reprezentují internetové služby a ty zbylé slouží pro záznam přihlašovacích údajů uživatelů.

V posledních kapitolách se nachází hodnocení celého systému, jeho využití v praxi a možnosti jak ho lze rozšiřovat. Součástí je také ukázka jak celý autentizační systém funguje. Systémy jednotného přihlašování jsou velmi rozšířené a jsou využívané mnoha světovými organizacemi ve veřejném i soukromém sektoru. Tématem systému jednotného přihlašování se zabývalo několik závěrečných prací napříč různými univerzitami. Převážně byly systémy postavené na jiné technologii než je SAML. V rámci práce Endreho Takácse na univerzitě VUT v Brně to byl protokol Kerberos. Tomu se také věnovala práce Bc. Jiřího Zifčáka z VŠB fakulty elektrotechniky a informatiky.

2 Princip systému jednotného přihlašování

Systém jednotného přihlašování neboli Single sign-on (SSO) je systém, který poskytuje uživateli možnost přístupu k více na sobě nezávislých aplikacím se stejnými přihlašovacími údaji. Plní funkci autentizace a autorizace uživatelů. Poskytovatelé internetových služeb (SP) k údajům přístup vůbec nemají, ty jsou uložené v jednom centrálním místě na straně poskytovatele identit (IdP). Ten také zajišťuje jednotnou správu osobních údajů, aby je nebylo při registraci u dalších služeb nutné vyplňovat znovu. Uživatel je v rámci jedné relace poté přihlášený ke vše ostatním službám spojených s tímto SSO systémem, ke kterým má přístupová práva. Při přechodu mezi internetovými službami není autentizace vyžadovaná znovu.



Obrázek 1: Princi SSO systému

2.1 Výhody SSO

Mezi hlavní výhody patří komfort při využívání internetu. To je hlavně díky tomu, že uživatel nemusí zadávat přihlašovací údaje pro jednotlivé služby. SSO snižuje čas strávený opětovným zadáváním hesla. Další z výhod je to, že služby poskytující jednotné přihlašování mají většinou silnější metody autentizace, například s využitím osobního certifikátu. Náklady vzniklé provozováním uživatelské podpory jsou také nižší, protože se nemusí tolik řešit ztráty hesel.

2.2 Rizika SSO

Využití jednoho hesla pro více internetových služeb je výhodou, ale na druhou stranu sebou nese bezpečnostní riziko. Pro útočníka je daleko jednodušší získat přístup k více informacím uživatele prolomením pouze jednoho hesla.

2.3 Různé metody implementace SSO

2.3.1 Metoda založená na federace identit

Tato část popisuje technologie na principu federace identit, které využívají protokoly SAML, OpenID a OAuth. Federovaný SSO systém je založený na důvěrném vztahu poskytovatele služeb a identit, aby zajistili uživateli přístup k zabezpečené webové službě nebo aplikaci.

2.3.2 OpenID

OpenID je otevřený standard sponzorovaný společnostmi Facebook, Microsoft, Google, PayPal a dalšími. Funkce je obdobná jako u SAML, zahrnuje tři základní prvky. Uživatel, kterému je poskytována identita, služba vyžadující identifikaci uživatele a OpenID poskytovatel, který identitu ověřuje.

2.3.3 OAuth

Open Authorization je otevřený standard, který na rozdíl od OpenID a SAML slouží pouze k autorizaci. OAuth je zaměřený na autorizační proces pro webové, desktopové a mobilní aplikace. Systém funguje na principu distribuce autorizačního tokenu.

2.3.4 SAML

Standard vytvořený v roce 2001 společností OASIS založený na XML pro výměnu autentizačních a autorizačních informací mezi skupinami. SAML je podrobněji popsán v kapitole č. 2.4.

2.3.5 Metoda založená na protokolu Kerberos

Kerberos je autentizační protokol, který umožňuje uživateli se přihlásit do Windows účtu a získat tak přístup k vnitřním aplikacím. Kerberos potřebuje, aby uživatel měl přístup k hlavnímu KDC, tj. Centrum pro distribuci klíčů. Autentizací poté uživatel získá od KDC zašifrovaný *ticket* pro danou službu, kterou chce používat.

2.3.6 Metoda založená na biometrii

Současné autentizace založené na biometrii mohou být pro zvýšení bezpečnosti a usnadnění procesu spojeny s SSO technologiemi. Tato autentizace může zahrnovat využití otisku prstů, skenování obličeje, sítnice nebo dokonce DNA.

2.3.7 Metoda založená na SmartCard

SSO pomocí čipových karet SmartCard umožňuje se uživateli autentizovat vložením této karty do čtečky a přihlásit se k dané aplikaci. Použitím informací z této karty, potom umožňuje přístup i k ostatním aplikacím.

2.3.8 Metoda založená na Cookie souborech

Tato metoda využívá soubory Cookie v prohlížeči k přenosu informací z prohlížeče na server bez toho, aby se uživatel musel znovu přihlašovat. Přihlašovací údaje jsou na straně uživatele zaznamenány a šifrovány před tím, než jsou vloženy do cookie a odeslány na server. Server potom tyto údaje dešifruje a ověří vůči databázi uživatelů.

2.4 Standard SAML

Standard SAML neboli Security Assertion Markup Language je framework založený na XML, který popisuje a umožňuje výměnu zabezpečených dat, konkrétně autentizačních a autorizačních informací, mezi jednotlivými poskytovateli online služeb. Tyto zabezpečené informace jsou vyjádřeny ve formě SAML tvrzení, kterým aplikace mohou důvěřovat. Tento standard přesně definuje syntaxi a pravidla pro dotazování, vytváření, komunikaci těchto SAML tvrzení.

Standard byl vyvinutý organizací OASIS, která ho také spravuje. Jedná se o globalní neziskové konsorcium věnující se vývoji a nasazení otevřených standardů v oblasti bezpečnosti, internetu věcí, a další oblastí.

Hlavní důvody nasazení SAMLu pro výměnu bezpečnostních informací:

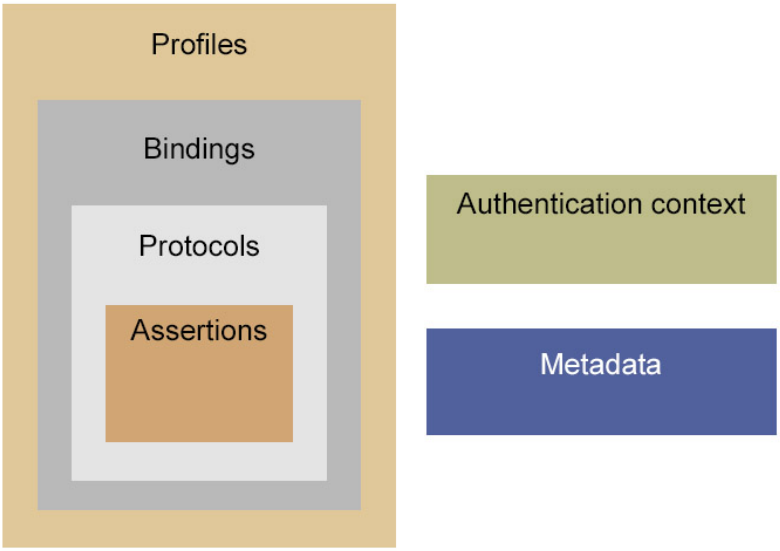
- **Single Sign-On:** SAML sjednocuje standard pro výměnu autentizačních informací pro systémy jednotného přihlašování.
- **Společná identita:** Uživatel má stejnou identitu napříč různými službami, což umožňuje lepší správu a uchovávání informací o uživateli.
- **Webové služby a ostatní standardy:** SAML dovoluje díky svému formátu a modularitě využití i v jiných webových službách a frameworcích, které nejsou založené na SAMLu.

Účastníci podílející se na výměně informací v rámci SAML standardu jsou SAML *asserting party* a SAML *relying party*, ve většině případů se jako účastník bere také uživatel.

SAML *asserting party* je systémová entita, která vytváří SAML tvrzení. Někdy se nazývá jako SAML *authority*. Důležitý typ autority je **identity provider**. SAML *relying party* je systémová entita, která přijímá informace od jiné entity. Typicky jde o autoritou vydané tvrzení. Této entitě se říká **service provider**.

2.4.1 SAML architektura

Základní koncept SAML architektury je složen z komponent poskytující přenos identity, autentizace, atributů a autorizačních informací mezi anonymními organizacemi, které mají mezi sebou důvěryhodný vztah. Definuje strukturu a obsah tvrzení a protokolových zpráv použitých k přenosu informací.



Obrázek 2: Základní koncept SAML architektury

Profiles je kombinace SAML *assertions, protocols a bindings* pro podporu určitých případů užití, jako například Web Browser SSO Profile. Profily typicky definují omezení obsahu těchto částí.

Bindings převádí SAML protocols zprávy na standardní komunikační protokoly jako jsou HTTP nebo SOAP.

Protocols spravuje dotazy a odpovědi pro získání SAML tvrzení(assertions), a zajišťuje správu identit.

Assertions neboli tvrzení je dokument s bezpečnostními informacemi. Patří zde informace o autentizaci, atributech a rozhodnutí o autorizaci.

Metadata XML dokument, který obsahuje informace o poskytovatelích identit a služeb. Zajišťují bezpečný přenost mezi těmito subjekty.

Authentication context je používaný u autentizace k tomu, aby držel detailní informace o typu, síle a dalších informací o autentizaci.

2.4.2 Výměna informací mezi IdP a SP

Tato sekce je zaměřená na výměnu zpráv mezi poskytovatelem identit a poskytovatelem služeb s využitím Web Browser SSO profilu v rámci SAML V2.0.

Web Browser SSO profile nabízí široké možnosti využití, první z nich je výměna zpráv v závislosti na tom jestli je komunikace iniciovaná ze strany SP nebo IdP. Druhá je založená na tom jaké SAML *Bindings* jsou použity pro výměnu zpráv mezi SP a IdP.

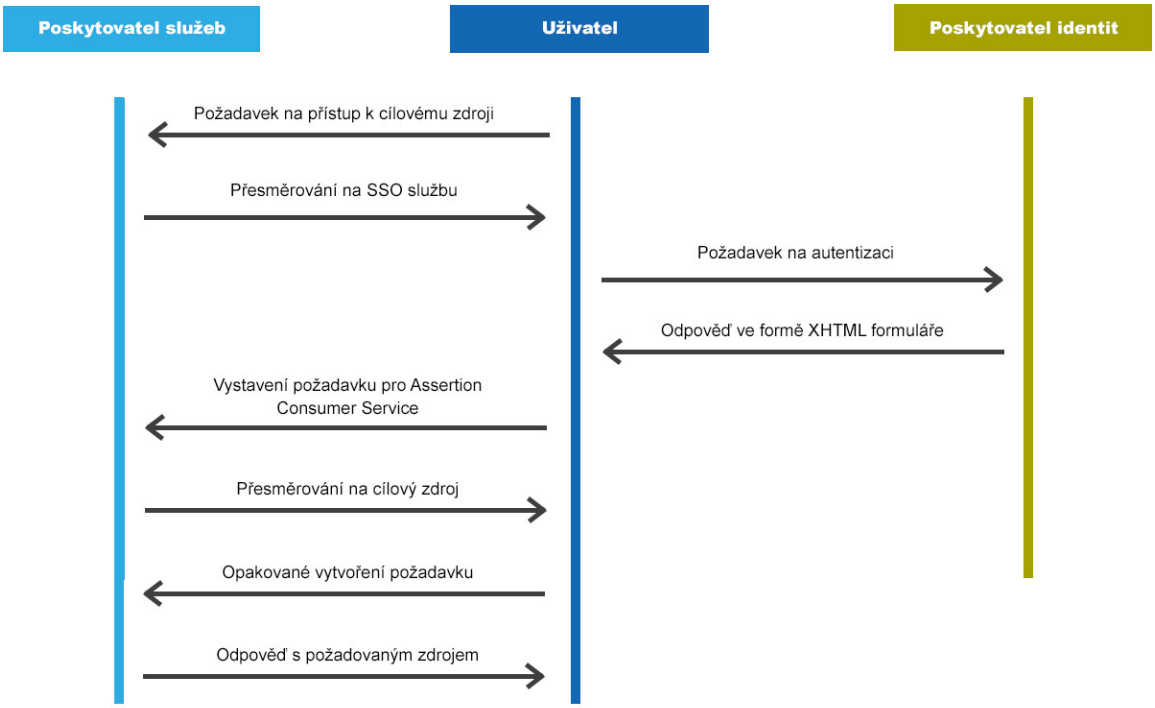
Mezi nejběžnější scénáře patří komunikace vyvolaná ze strany poskytovatelem služeb, kdy se uživatel pokouší získat přístup v rámci dané služby. Nicméně veškeré přihlášení je řešeno na straně poskytovatele identit, kde je uživatel přesměrován.

2.4.3 Ukázka výměny zpráv iniciovaná SP s využitím redirect/POST binding

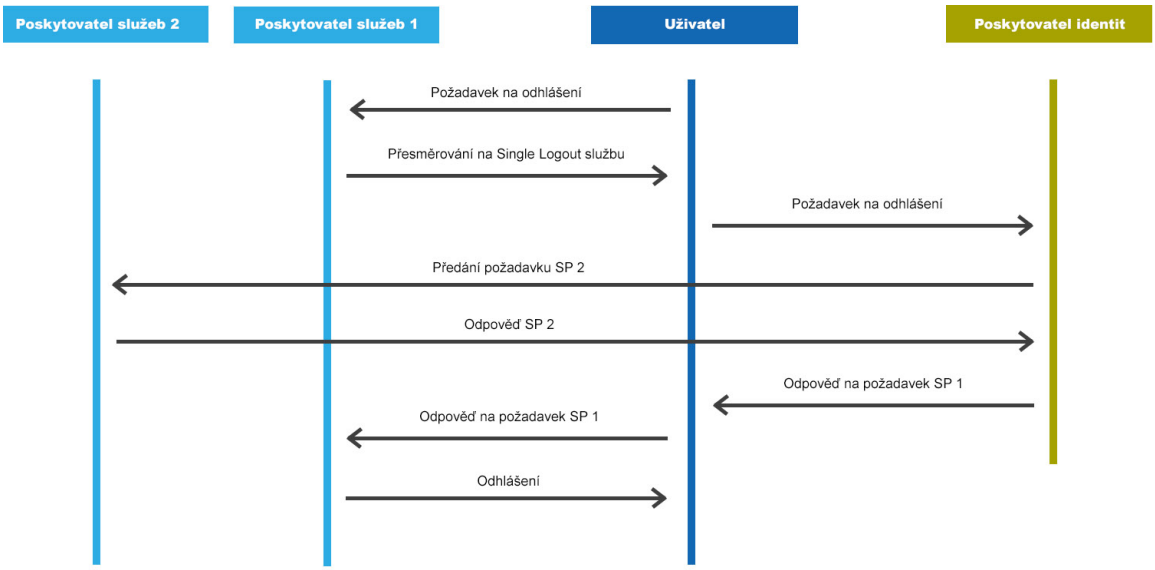
1. **Požadavek na straně SP**: Uživatel zažádá o přístup k cílovému zdroji u SP. SP provede kontrolu jestli už existuje přihlášení a pokud ano následuje bod 8.
2. **Přesměrování na IdP SSO službu**: SP vygeneruje požadavek, který prohlížeč přesměruje na příslušnou IdP SSO službu.
3. **Předání požadavku na autentizaci SSO službě na straně IdP**
4. **Zaslání odpovědi ve formě XHTML formuláře**: SSO služba ověří správnost požadavku a odešle odpověď v podobě XHTML formuláře.
5. **Vystavení požadavku pro Assertion Consumer Service na straně SP**
6. **Přesměrování na cílový zdroj**: Assertion Consumer Servis zpracuje odpověď a přihlásí uživatele, následně ho přesměruje.
7. **Opakované vytvoření požadavku na straně SP**
8. **Odpověď s požadovaným zdrojem**: Ověří, že je uživatel přihlášený a poskytne mu cílové informace.

2.4.4 Výměna zpráv iniciovaná SP pro odhlášení

1. **Požadavek na straně SP**: Vytvoření globálního požadavku na odhlášení.
2. **Přesměrování na IdP Single Logout službu**: SP odstraní lokální přihlášení a následně odešle informace o odhlášení IdP.
3. **Předání požadavku druhému SP**: IdP pošle požadavek na odhlášení všem ostatním účastníkům, to znamená druhému SP.
4. **Odpověď druhého SP**
5. **Odpověď na požadavek prvního SP**
6. **Odhlášení**: První SP informuje uživatele, že byl odhlášen ze všech ostatních SP.



Obrázek 3: Diagram komunikace iniciované SP s využitím request/POST binding



Obrázek 4: Diagram komunikace globálního odhlášení

2.5 Použité nástroje a protokoly

2.5.1 Ubuntu 18.04

Ubuntu je Linuxová distribuce s otevřeným zdrojovým kódem, která běží na mnoha zařízeních od cloudu, přes osobní počítače, smartphony až po nejmenší zařízení jako jsou roboti, drony nebo ovladače chytré domácnosti. Je založená na Debianu a oficiálně vydávaná ve třech verzích, pro PC, servery a základní pro zařízení internetu věcí a roboty. Kterákoliv z těchto verzí může běžet samostatně na počítači, nebo virtuálním stroji. Ubuntu je vyvíjeno společností Canonical a komunitou dalších vývojářů, kteří této distribuci zajišťují dlohodobou podporu.

2.5.2 LXD/LXC

Technologie linuxových kontejnerů byla vyvinuta před dlouhou dobou a jedná se o virtualizační technologii, která funguje na úrovni operačního systému. Díky ní můžeme vytvářet a provozovat více na sobě nezávislých virtuálních linuxových prostředí na jednom Linuxovém stroji. LXC převážně využívá funkcionality *cgroups* a *namespaces* představené ve verzi kernelu 2.6.24.

Mezi důvody využití kontejnerů patří to, že mají vlastí izolovaný souborový systém. To znamená, že software běžící v kontejneru nijak nekoliduje s tím, který běží mimo něj. Další výhodou je to, že uvnitř kontejneru lze provozovat různé Linuxové distribuce. To přináší všechny balíčky, které daná distribuce obsahuje.

Hlavním rozdílem mezi LXC a LXD je to, že LXC je původní starší verze pro správu kontejnerů, ale je stále podporovaná. LXD umožňuje nový přístup ke správě kontejnerů využívající LXC příkazy.

2.5.3 SSH

Secure Shell je protokol pro zabezpečení komunikace přes nezabezpečenou síť. SSH byl navržený jako náhrada za Telnet. Zabezpečuje autentizaci účastníků komunikace, transparentní šifrování přenášených dat, zajišťuje integritu a kompresi. Pro autentizaci uživatelů využívá asymetrické šifrování.

SSH se používá při práci na vzdáleném serveru. Klient se připojuje na *SSH daemon*, který rozhoduje o přijetí spojení a požadované autentizaci.

2.5.4 Apache 2

Apache HTTP Server je volně dostupný software pro provoz webového serveru, který byl vyvinutý otevřenou komunitou vývojářů po Apache Software Foundation. Velká většina Apache HTTP Serverů běží pod operačním systémem Linux.

Apache podporuje různé vylepšení, hodně z nich je implementováno jako moduly, které rozšiřují základní funkce. Patří sem autentizační moduly zahrnující podporu SSL a TLS. Dále podpora programovacích jazyků na straně serveru například PHP, Python, Perl. Apache dovoluje

vytvoření virtuálního hostingu. Jeden Apache webový server potom poskytuje více webových stránek najednou. Apache má také podporu autentizace pomocí hesla a digitálního certifikátu.

Protože zdrojový kód je volně dostupný, kdokoli může uzpůsobit server svým potřebám díky různým přídavným addonům.

2.5.5 PHP 7.3

PHP je skriptovací programovací jazyk, běžící na straně serveru. Umožňuje ukládat a měnit data webových stránek. Pokud se PHP používá pro dynamické internetové stránky, tak se veškeré skripty provádějí na straně serveru a k uživateli se přenášejí pouze jejich výsledky. Syntaxe jazyka PHP je inspirovaná programovacími jazyky jako Perl, C, Pascal nebo Java. PHP není závislý na platformě, to znamená, že lze kód přenášet poměrně jednoduše mezi různými operačními systémy, bez nutnosti provádět složité úpravy. PHP podporuje různé knihovny od těch pro zpracování textu, grafiky až po ty pro přístup k databázovým systémům. PHP také umožňuje podporu celé řady protokolů jako HTTP, SMTP, FTP, LDAP a další.

2.5.6 SSL

Secure Security Layer je protokol pracující mezi transportní a aplikační vrstvou, který zajišťuje zabezpečenou komunikaci šifrováním a autentizací komunikujících stran.

SSL se využívá nejčastěji k zabezpečení komunikace s webovými servery pomocí HTTPS, kdy je spojení šifrované. SSL využívá asymetrického šifrování. Každá komunikující strana má dva klíče, veřejný a soukromý. Veřejným klíčem protistrany se zpráva zašifruje a následně protistrana svým soukromým tuto zprávu dešifruje. SSL pro toto využívá digitálních certifikátů.

2.5.7 SimpleSAMLphp

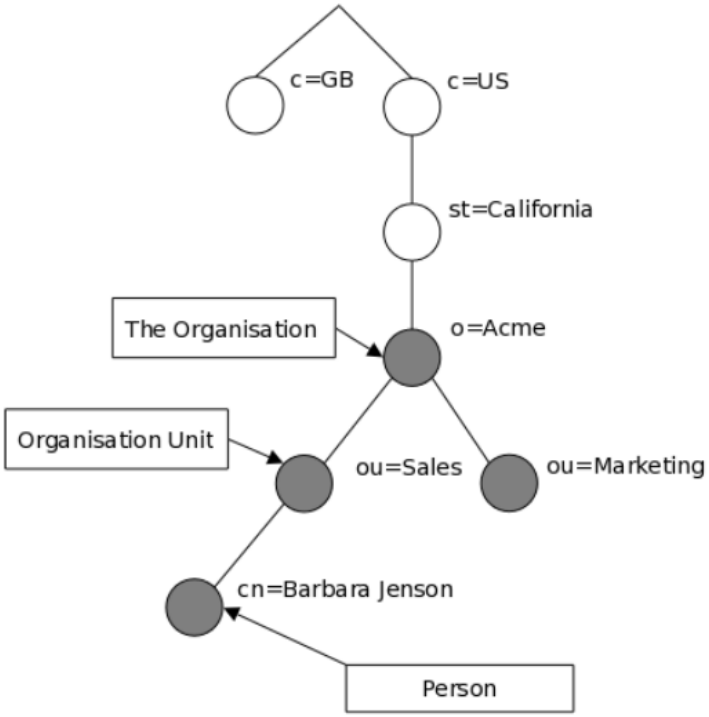
SimpleSAMLphp je knihovna napsaná v PHP řešící autentizaci. Hlavně se zaměřuje na poskytnutí podpory pro SAML 2.0 poskytovatele služeb a poskytovatele identity. Nicméně podporuje také i jiné protokoly a frameworky jako třeba OAuth, OpenID nebo Shibboleth 1.3. Knihovna je jednoduše rozšiřitelná pro vývoj vlastních modulů.

2.5.8 LDAP

Lightweight Directory Access Protocol, jedná se o adresářovou informační službu. To znamená, že umí uchovávat záznamy, jako jsou například seznam lidí firmy, přihlašovací jména, domovské adresáře, osobní informace, jména emailů nebo čísla telefonů. LDAP může stejně tak dobře uchovávat nastavení uživatelských programů. Data se nemusejí nutně vztahovat na osoby, protokol může pomoct vyhledat různé přístroje ve firemní síti a zobrazit jejich umístění.

LDAP je jednoduchý a dobře navržený protokol umožňující nejen klást poměrně složité dotazy, ale i vkládat, modifikovat a mazat záznamy. Celou službu je možno si představit jako

veliký strom či adresář založený na souborovém systému. Tento strom obsahuje záznamy (entries). Každý záznam musí mít definované povinné a volitelné atributy (attributes). Ty definujeme pomocí objektů (object class), které jsou nastaveny na serveru. Standardně je nadefinováno jen několik základních objektů typu person nebo organization. Další je třeba si definovat samostatně, v závislosti na aplikace, která to vyžaduje.



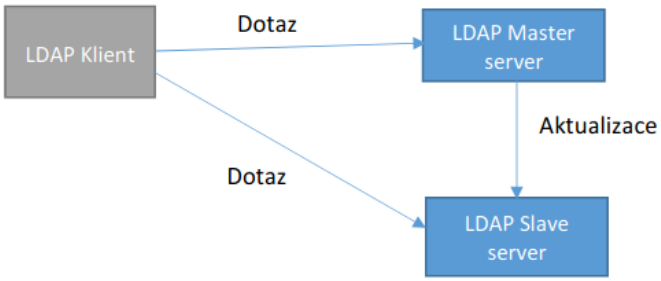
Obrázek 5: Ukázka adresářové stromové struktury LDAP

2.5.9 OpenLDAP

OpenLDAP je opensource řešení LDAP, postavené na nerelační databázi, která je nastavená tak, aby se k datům mohlo přistupovat, ale ne je zapisovat. Nejčastěji se používá k autentizaci uživatelů.

2.5.10 Master-Slave mód

LDAP servery, které fungují v modelu mater-slave slouží k obsluze více stejných dotazů. LDAP master server aktualizuje adresáře LDAP slave serveru, slave server poté obsluhuje dotazy ke čtení a chová se jako záložní. Základní nastavení LDAP serveru je v módu master, při čemž serverů v módu slave může být jeden a více.



Obrázek 6: Schema dvou LDAP serverů v módu Master-Slave

2.5.11 Nette framework

Volně dostupný framework pro vytváření webových aplikaci v PHP a zjednodušení jejich tvorby. Jeho autorem je český vývojář David Grudl. Framework je dobře objektově navržený a v Čechách velmi rozšířený. Mezi projekty, které ho využívají patří například GE Money, Slevomat, ČSFD a velké množství různých eshopů.

Framework je ucelených soubor knihoven. K důvodům proč využívat právě Nette, patří to, že spousta důležitých funkcí v PHP chybí nebo se s nimi špatně pracuje. Tyhle mezery řeší právě Nette. Dalším z důvodů je úspora času. Použitím vhodných knihoven ušetříte čas, ale také bude aplikace daleko lépe udržovatelná a její tvorba přehlednější.

Nette je MVC framework, což znamená, že aplikace stojí na třech typech komponent, které řeší řízení, logiku a výstup. Mezi komponenty patří *Presentery*, které se starají o řízení a zprostředkování komunikace mezi uživatelem a *Modely*. Ty obsahují logickou část aplikace jako jsou výpočty a práci s databází. Poslední částí jsou *Pohledy*, které zajišťují výstup v podobě HTML kódu. Pro jejich vytváření Nette využívá vlastní šablonovací jazyk Latte, který umožňuje do nich vkládat data z PHP pomocí speciálních značek.

3 Návrh řešení a konfigurace SSO systému

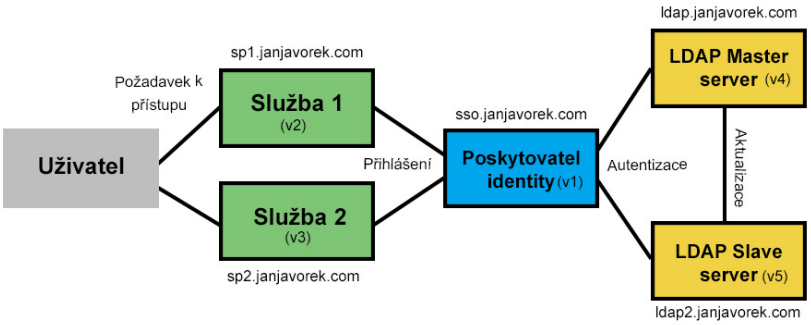
Pro požadovanou funkci systému bylo nutné vytvořit vhodné prostředí. To spočívá v několika komunikujících serverech. Nejvhodnějším řešením bylo využití virtualizace. Jedná se o přidělení systémových prostředků daného stroje konkrétnímu virtuálnímu stroji, ten se pak chová jako samostatný systém. Takto lze provádět virtualizaci velmi jednoduše. Pro SSO systém byly použity konkrétně LXD/LXC kontejnery a to z důvodů jejich vzájemné nezávislosti včetně nezávislosti na operačním systému. Jako operační systém pro tvorbu jsem zvolil Linuxovou distribuci Ubuntu z důvodu toho, že nejlépe využívá zdroje při tvorbě více kontejnerů. Řešení spočívá ve vytvoření pěti kontejnerů, kdy jeden se chová jako poskytovatel identit, další dva jako poskytovatelé internetových služeb a ty poslední jako LDAP servery s databází údajů o uživatelích.

Tabulka 1: Přehled IP adres a domén použitých Linuxových kontejnerů

Název	Doména	IPv4, IPv6 adresy
v1	sso.janjavorek.com	10.71.122.131 fd42:dea0:d7f5:ef23:216:3eff:fe49:70e
v2	sp1.janjavorek.com	10.71.122.24 fd42:dea0:d7f5:ef23:216:3eff:fe34:80c2
v3	sp2.janjavorek.com	10.71.122.98 fd42:dea0:d7f5:ef23:216:3eff:fe87:8bd0
v4	ldap.janjavorek.com	10.71.122.68 fd42:dea0:d7f5:ef23:216:3eff:fe2c:8ce0
v5	ldap2.janjavorek.com	10.71.122.46 fd42:dea0:d7f5:ef23:216:3eff:fe8f:4b77

Tabulka 2: Přehled funkcí použitých Linuxových kontejnerů

Název	Funkce	Potřebné protokoly a nástroje
v1	Poskytovatel identit	Apache 2, PHP, SimpleSAMLphp, SSH
v2	Poskytovatel služeb	Apache 2, PHP, SimpleSAMLphp, SSH, Nette
v3	Poskytovatel služeb	Apache 2, PHP, SimpleSAMLphp, SSH, Nette
v4	LDAP Master server	OpenLDAP, SSH
v5	LDAP Slave server	OpenLDAP, SSH

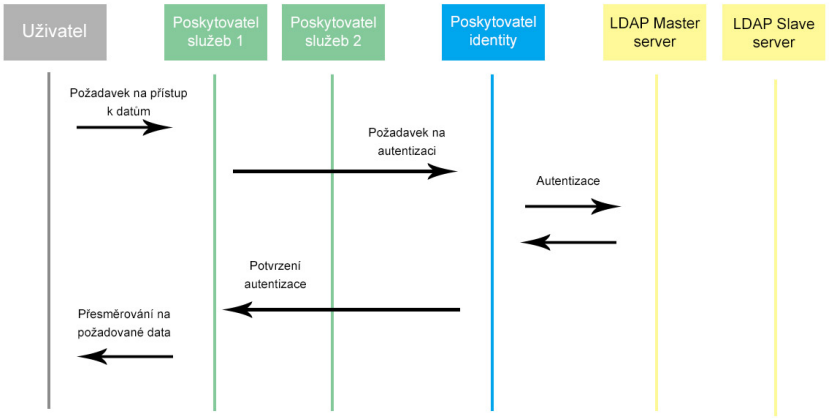


Obrázek 7: Schéma rozložení serverů SSO systému

Pro implementaci řešení SSO systému prostřednictvím SAML jsem zvolil knihovnu simpleSAMLphp. Knihovna nabízí elegantní řešení implementace poskytovatelů služeb i identity včetně napojení různých adresářových systému pro správu uživatelů. Také jí je možné rozšiřovat o vlastní moduly. Poskytovatel identity reprezentuje autoritu, u které se potom uživatelé přistupující k různým službám ověřují. Sám poté uživatele autentizuje, pomocí údajů o nich uložených v adresářové struktuře LDAP serveru.

LDAP servery jsou vytvořeny v rámci nástroje OpenLDAP a nastaveny v módu Master-Slave. Tento mód umožňuje spravovat uživatele skrze Master server, kdy Slave server pouze aktualizuje svou adresářovou strukturu podle Master serveru. Toto řešení je vhodné hlavně, když hlavní z LDAP server je mimo provoz nebo je přetížený. Autentizaci uživatelů poté provádí druhý LDAP server.

Internetové služby jsou potom reprezentovány webovými aplikacemi, které jsou vytvořeny pomocí frameworku Nette. V aplikaci je možnost se přihlásit, resp. dát požadavek na přihlášení, a následně po úspěšné autentizaci vidět údaje o přihlášeném uživateli. Aplikace je připravená k rozšíření.



Obrázek 8: Ukázka komunikace systému SSO podle návrhu

3.1 Postup konfigurace SSO

V této kapitole je popsán postup konfigurace všech nástrojů potřebných k provozu SSO systému na standardu SAML.

3.2 Konfigurace LXD/LXC kontejnerů

Pro účely systému bylo třeba nainstalovat nástroj pro tvorbu kontejnerů LXD, následně ho nastavit a poté vytvořit 5 kontejnerů, které budou běžet na operační systém Ubuntu 18.04. Konkrétně se vytvoří kontejnery v1, v2, v3, v4, v5.

```
// Instalace LXD nástroje
$ sudo apt update
$ sudo apt install lxd -y

//Inicializace LXD
$ sudo lxc init

// Vytvoření kontejnerů, X je nahrazeno číslem kontejneru
$ sudo lxc launch ubuntu:18.04 vX

// Přístup ke kontejnerům, X je nahrazeno číslem kontejneru
$ sudo lxc exec vX bash
```

Kontejnery jsou vytvořené a spuštěné. Aktuálně je možné ke kontejnerům přistoupit pouze skrze Bash. Podrobnosti jako funkce kontejnerů, jejich IP a další jsou v tabulce č.1 a č. 2.

3.2.1 Nastavení SSH přístupu

Pro jednodušší a bezpečnější přístup ke kontejnerům je vhodné nastavit SSH připojení. Je třeba nainstalovat nástroj OpenSSH, přidat autentizaci pomocí veřejného klíče a povolit v nastavení firewallu SSH port.

Také je třeba v kontejneru vytvořit uživatele a nastavit mu práva. Pro lepší orientaci je vhodné nastavit také lokální mapování doménových jmen pro IP adresy.

```
//Vytvoření uživatele v kontejneru se jménem itsme
$ adduser itsme

//Nastavení práv
$ usermod -aG sudo itsme
```

Provedení změn v souboru */etc/hosts*

```
#v1
10.71.122.131 sso.janjavorek.com www.sso.janjavorek.com
#v2
10.71.122.131 sp1.janjavorek.com www.sp1.janjavorek.com
#v3
10.71.122.131 sp2.janjavorek.com www.sp2.janjavorek.com
#v4
10.71.122.68 ldap.janjavorek.com
#v5
10.71.122.46 ldap2.janjavorek.com
```

```
//Instalace OpenSSH
$ sudo apt update
$ sudo apt install openssh-server

//Autentizace
//Vytvoření adresáře pro SSH klíče
$ mkdir -p ~/.ssh
$ chmod 700 ~/.ssh

//Vygenerování klíčů
$ ssh-keygen -t rsa

//Zkopírování veřejného klíče do kontejneru, X je
//nahrazeno lokální doménou např. sso.janjavorek.com
$ ssh-copy-id -i ~/.ssh/id_rsa.pub itsme@X

//Povolení SSH port ve firewallu
$ sudo ufw allow OpenSSH
$ sudo ufw enable
```

Nyní je možné provést připojení ke každému kontejneru pomocí SSH.

```
//Připojení přes SSH, X je nahrazeno lokální doménou kontejneru ke kterému se p
    řipojujete např. sso.janjavorek.com
$ ssh itsme@X
```

3.3 Konfigurace Apache 2

Další podstatnou částí je instalace a nastavení webového serveru Apache. Provedeme instalaci, dále nastavíme ServerName a povolíme provoz ve firewallu. Celou tuto část je nutné provést pro oba servery poskytovatelů služeb a poskytovatele identity.

```
//Instalace Apache2
$ sudo apt update
$ sudo apt install apache2
```

Do souboru */etc/apache2/apache2.conf* je třeba přidat doménové jméno serveru nebo IP. Jméno bude podle aktuálního virtuálního stroje, pro který webové server nastavujeme, např. *sso.janjavorek.com*.

```
...
ServerName domena_nebo_IP
```

```
//Test konfigurace a restart apache2 serveru
$ sudo apache2ctl configtest

$ sudo systemctl restart apache2
```

Povolení Apache v bráně firewall.

```
//Kontrola zda firewall dovoluje HTTP,HTTPS provoz
//(porty 80,443)
$ sudo ufw app info "Apache Full"

//Povolení ve firewallu
$ sudo ufw allow in "Apache Full"
```

3.3.1 Nastavení SSL certifikátu

K tomu abychom zajistili bezpečný přenos dat mezi jednotlivými službami, uživatelem a poskytovatelem identity slouží SSL. SSL certifikát nejprve pomocí OpenSSL vytvoříme a následně nastavíme webový server tak, aby ho využíval.

Certifikát, který vytváříme bude podepsaný sám sebou, ve formátu X.509 s dobou platnosti 1 rok a šifrovaný RSA-2048.

```
//Vytvoření certifikátu
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/ssl/private/apache-selfsigned.key
-out /etc/ssl/certs/apache-selfsigned.crt
```

Nyní máme vytvořený klíč a certifikát podepsaný tímto klíčem, teď je nutné upravit soubor pro Apache Virtual Host.

```
//Zálohování původního souboru
$ sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-
    available/default-ssl.conf.bak
```

V souboru */etc/apache2/sites-available/default-ssl.conf* upravíme položky ServerAdmin, ServerName a cesty k certifikátu a klíči. Po úpravách by měl soubor vypadat následovně.

```
<IfModule mod_ssl.c>
   <VirtualHost _default_:443>
      ServerAdmin admin@sso.janjavorek.com
      ServerName sso.janjavorek.com

      DocumentRoot /var/www/html

      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined

      SSLEngine on

      SSLCertificateFile    /etc/ssl/certs/apache-selfsigned.crt
      SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

      <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
      </FilesMatch>
      <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
      </Directory>

   </VirtualHost>
</IfModule>
```

```
//Zapnutí ssl modu v Apache
$ sudo a2enmod ssl
$ sudo a2enmod headers

$ sudo a2ensite default-ssl
```

```
//Kontrola chyb v syntaxi
$ sudo apache2ctl configtest

//Restart
$ sudo systemctl restart apache2
```

3.4 Konfigurace PHP 7.3

Celá konfigurace PHP spočívá v instalaci následujících balíčků a úpravě souboru */etc/apache2/mods-enabled/dir.conf*.

```
//Instalace
$ sudo apt install php libapache2-mod-php php-mcrypt php-mysql php-xml php-
    mbstring php-curl php-memcache php-ldap memcached

$ sudo systemctl restart apache2
```

V souboru */etc/apache2/mods-enabled/dir.conf* upravíme následující.

```
<IfModule mod_dir.c>
   DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

Nyní máme vše připravené k implementaci samotného SimpleSAMLphp systému.

3.5 Konfigurace SimpleSAMphp

Instalace SimpleSAMLphp zahrnuje několik kroků. Musíme stáhnout samotný software a všechny jeho komponenty. Následně je třeba také udělat změny v nastavení webového serveru Apache. Tato část je stejná jak pro konfiguraci poskytovatele služeb, tak pro poskytovatele identity.

Stažení nejnovější verze SimpleSAMLphp ze stránek projektu. Následně je třeba extrahovat obsah staženého souboru a vše v adresáři *simplesamlphp-1.x.y*(x,y představují číslo aktuální verze) zkopírovat do adresáře */var/simplesamlphp*.

```
//Stažení SimpleSAMLphp
$ wget https://simplesamlphp.org/download?latest

//Rozbalení souboru
$ tar zxf download?latest
```

```
//Zkopírování souboru, x,y je třeba nahradit aktuální verzí
$ sudo cp -a simplesamlphp-1.x.y/. /var/simplesamlphp/
```

Nastavení Apache aby využíval SimpleSAMLphp a pracoval s protokolem HTTPS, který zabezpečuje přenos dat. Jediný adresář viditelý pro web musí být */var/simplesamlphp/www*, proto je potřeba upravit v nastavení Apache *VirtualHost*.

```
//Úprava souboru, DOMAIN se nahradí doménou aktuálního //serveru např. sso.
    janjavorek.com
$ sudo nano /etc/apache2/sites-available/DOMAIN-ssl.conf
```

Obsah souboru by měl vypadat po úpravách následovně.

```
<IfModule mod_ssl.c>
   <VirtualHost _default_:443>
   ServerAdmin admin@sso.janjavorek.com
   ServerName sso.janjavorek.com

   DocumentRoot /var/www/html

   <Directory /var/simplesamlphp/www/>
      Require all granted
   </Directory>

   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined

   SSLEngine on

   SSLCertificateFile    /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

   <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
   </FilesMatch>
   <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
   </Directory>
   </VirtualHost>
</IfModule>
```

Dále je potřeba nastavit v tomto souboru *Alias*, aby docházelo k přesměrování na příslušný adresář a SimpleSAMLphp mohl řídit celý proces. Následně je třeba dát adresáři */var/simplesamlphp/www* přístupová práva. Tohle umožní službě SimpleSAMLphp přístup skrze celý web.

```
...

   Alias /simplesaml /var/simplesamlphp/www

...
```

Na závěr webový server restartujeme pro aplikování změn.

```
$ sudo systemctl restart apache2
```

Konfigurace samotného SimpleSAMLphp spočívá v úpravách souboru *config.php*

```
//Otevření souboru config.conf
$ nano /var/simplesamlphp/config/config.conf
```

Je nutné nastavit administrátorské heslo prostřednictvím položky *'auth.adminpassword'*. Toto heslo umožňuje přístup k určitým stránkám, které slouží k nastavení v SimpleSAMLphp rozhraní.

```
...

'auth.adminpassword'  => 'heslo'

...
```

Jako další položku je třeba nastavit tzv. *secret salt*. Jedná se nejlépe o náhodně vygenerovaný řetězec znaků. Některé části SimpleSAMLphp používají tento řetězec pro vytvoření bezpečostních hašů. Pokud bude tato položka nezměněna bude docházet k chybám.

Pro vytvoření řetězce lze použít OpenSSL *rand* funkci. Přepínač *-base64 32* zajistí, že řetězec bude kódovaný Base64 a dlouhý 32 znaků. V novém terminálu, po přípojení na server se spustí následující příkaz.

```
//Vygenerování náhodného řetězce
$ openssl rand -base64 32
```

Potom se v konfiguračním souboru změní položka *'secretsalt'* z původního řetězce na ten vygenerovaný.

```
...

'secretsalt' => 'vygenerovany_retezec'

...
```

Jako další část se nastaví kontaktní údaje pro technickou podporu. Tyto informace budou potom dostupné ve vygenerovaných metadatech a SimpleSAMLphp bude na uvedenou emailovou adresu odesílat automaticky vytvořené chybové hlášení. Údaje se nastaví ve stejném souboru skrze položky *'technicalcontact_name'* a *'technicalcontact_email'*.

```
...

'technicalcontact_name' => 'Administrator'
'technicalcontact_email' => 'admin@sso.janjavorek.com'

...
```

Dále se nastaví příslušné časové pásmo v sekci *'timezone'*.

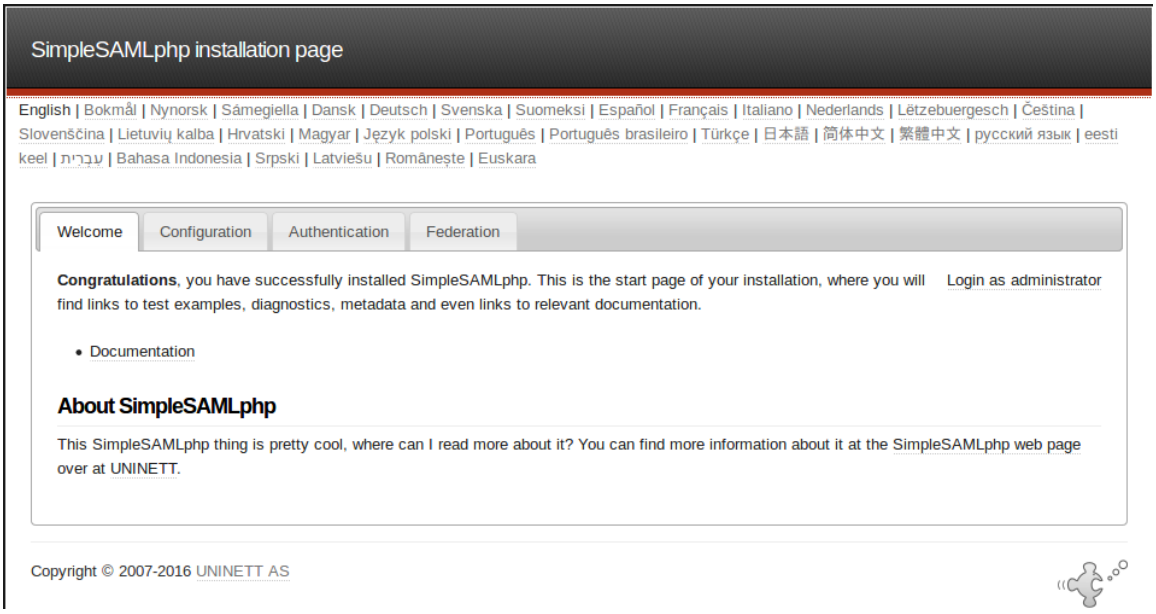
```
...

'timezone' => 'Europe/Prague'

...
```

Po uložení a zavření tohoto souboru je dostupná stránka *https://sso.janjavorek.com/simplesaml*. Je třeba se ujistit, že jsou nainstalovány všechny potřebné komponenty PHP. Po zvolení položky *Configuration* a přihlášení se jako administrátor, je vidět seznam potřebných PHP komponent i s tím jestli jsou nainstalované.

Ověřením funkčnosti přihlášení administrátora a PHP je dokončena část stejná pro poskytovatele služeb i identity. Dále bude popsána část specifická pro každého poskytovatele zvlášť.



Obrázek 9: SimpleSAMLphp rozhraní

3.5.1 Konfigurace IdP

Základní instalace a konfigurace je provedena, teď je třeba nastavit autentizační zdroj pro ověřování uživatelů. To spočívá v několika krocích. Zapnutí funkcionality poskytovatele identity, nastavení autentizačního modulu, self signed certifikátu, metadat a přidání informací o poskytovatelích služeb. Také je potřeba přidat informace o poskytovateli identity i poskytovateli služeb.

Prvním krokem, který je nutné udělat je zapnout funkcionalitu poskytovatele identity. To se provede úpravou souboru *config/config.php*, konkrétně volbou *'enable.saml20-idp'*.

```
...

'enable.saml20-idp' => true,

...
```

Dalším krokem je volba autentizačního modulu, který uživatele ověřuje. Existuje spousta modulů k ověřovaní v mém případě jsem zvolil **ldap:LDAP** resp. **ldap:LDAPMulti**. Uživatelé se tedy budou ověřovat skrze LDAP servery. Jejich podrobné nastavení je popsáno dále.

Modul **ldap:LDAPMulti** se používá v případě, že existuje více oddělených LDAP serverů, **ldap:LDAP** při využití jednoho. K jejich použití je potřeba provést úpravy v souboru *config/authsources.php* a přidat následující část.

```
...
/* janjavorek-ldap představuje název autentizačního zdroje*/

'name-ldap' => array(
   /* Modul */
   'ldap:LDAP',

   /* Hostname použitých LDAP serverů */
   'hostname' => 'ldap://ldap.janjavorek.com ldap://ldap2.janjavorek.com',

   /* Pattern se použije k vytvoření uživatelského DN */
   'dnpattern' => 'uid=%username%,ou=people,dc=ldap,dc=janjavorek,dc=com',
),
...
```

Nastavení metadat pro poskytovatele identit v souboru *metadata/samp20-idp-hosted.php*. Součástí je určení certifikátu a klíče, kterým IdP podepisuje SAML zprávy. SimpleSAMLphp pracuje pouze s RSA certifikáty. Také se nastaví příslušný autentizační zdroj ze souboru *config/authsource.php*

```
<?php

/* SAML 2.0 IdP configuration for SimpleSAMLphp. */

$metadata['__DYNAMIC:1__'] = [

   /* Hostname serveru používající tuto SAML entitu. */
   'host' => '__DEFAULT__',

   // X.509 klíč a certifikát.
   'privatekey' => 'apache-selfsigned.pem',
   'certificate' => 'apache-selfsigned.crt',

   /* Autentizační zdroj nastavený v souboru authsources.php*/
   'auth' => 'name-ldap',

   /* uri NameFormat pro attributy. */
   'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri$
```

```
    'authproc' => [

    /* Převod LDAP jmen na oids. */
    100 => ['class' => 'core:AttributeMap', 'name2oid'],
    ],
];
```

Přidání metadat všech poskytovatelů služeb pro IdP. IdP potřebuje vědět o všech SP, které se připojují. Nastavení se provádí v souboru *metadata/saml20-sp-remote.php*. Metadata je možné získat v rozhraní webu každého z SP a IdP po přihlášení jako administrátor v záložce federace.

Ukázka metadat pro prvního SP. Obdobně bude vypadat část pro další SP.

```
    $metadata['https://sp1.janjavorek.com/'] = array (
    'SingleLogoutService' => array (
    0 => array (
        'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
        'Location' => 'https://sp1.janjavorek.com/simplesaml/module.php/saml/sp/
            saml$
        ),
    ),
    'AssertionConsumerService' => array (
    0 => array (
        'index' => 0,
        'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST',
        'Location' => 'https://sp1.janjavorek.com/simplesaml/module.php/saml/sp/
            saml$
        ),
    1 => array (
        'index' => 1,
        'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:browser-post',
        'Location' => 'https://sp1.janjavorek.com/simplesaml/module.php/saml/sp/
            saml$
    ),
    2 => array (
        'index' => 2,
        'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact',
        'Location' => 'https://sp1.janjavorek.com/simplesaml/module.php/saml/sp/
            saml$
        ),
```

```
  3 => array (
     'index' => 3,
     'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:artifact-01',
     'Location' =>    'https://sp1.janjavorek.com/simplesaml/module.php/saml/sp
        /saml$
  ),
  ),
  'contacts' => array (
       0 => array (
          'emailAddress' => 'admin@sso.janjavorek.com',
          'contactType' => 'technical',
          'givenName' => 'Itsme',
       ),
  ),
```

3.5.2 Konfigurace SP

Pro úspěšnou konfiguraci poskytovatele služeb je potřeba nejprve provést úpravy v souboru *config/authsources.php*, přidat certifikát a klíč pro zabezpečení komunikace, nastavit informace o poskytovateli identity a jeho výchozí doménu.

SP je konfigurován skrze soubor *config/authsources.php* zvlášť pro každého SP.

```
<?php

$config = [
    'sp1' => [
      'saml:SP',
      'entityID' => 'https://sp1.janjavorek.com/',
      'idp' => 'https://sso.janjavorek.com/simplesaml/saml2/idp/metadata$
      ],
];
```

Další částí je předání informací o poskytovateli identity. Metadata pro IdP jsou uložena v souboru *metadata/saml20-idp-remote.php*. Metadata o IdP se generují automaticky ve webovém rozhraní po přihlášení jako administrator v záložce federace.

```
<?php

$metadata['https://sso.janjavorek.com/simplesaml/saml2/idp/metadata.php'] =
    array (
   'metadata-set' => 'saml20-idp-remote',
   'entityid' => 'https://sso.janjavorek.com/simplesaml/saml2/idp/metadata.php'
       ,

   'SingleSignOnService' => array (
   0 => array (
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
      'Location' => 'https://sso.janjavorek.com/simplesaml/saml2/idp/SSOService
          .php
      ),
   ),
   'SingleLogoutService' => array (
   0 => array (
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
      'Location' => 'https://sso.janjavorek.com/simplesaml/saml2/idp/
          SingleLogoutS$
      ),
   ),
   'certData' => 'MIID9TCCAt2gAwIBAgIUWLhJS7/z2AcCBlIm/WtDy/
       eyXlIwDQYJKoZIhvcNAQ$
   'NameIDFormat' => 'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',
   'contacts' => array (
   0 => array (
      'emailAddress' => 'admin@sso.janjavorek.com',
      'contactType' => 'technical',
      'givenName' => 'Itsme',
      ),
   ),
);
```

Část konfigurace SimpleSAMLphp je dokončena, zbývá provést nastavení LDAP serverů.

3.6 Konfigurace LDAP serverů

Instalace a nastavení LDAP serverů spočívá v instalaci nástroje OpenLDAP na oba servery. Následně se nastaví inicializační data pro vytvoření nového uživatele a uživatel se zanese do LDAP struktury. Poslední částí je nastavení replikace Master-Slave, která funguje tak, že druhý LDAP server označený jako Slave kopíruje veškerá data a změny z prvního Master serveru. To zajišťuje stále dostupné přihlášení do systému v případě nefunkčnosti jednoho z nich.

Provedeme instalaci z repozitáře. V jejím průběhu je nutné určit několik věcí. Jako první administrátorské heslo pro přihlášení do LDAP adresáře, dále zadáme, že nechceme vytvořit první databázi automaticky, jako další zadáme doménu, jméno organizace a vybereme formát databáze MDB. Jako poslední tři volby v instalaci nastavíme, že nechceme odebrat databázi v případě, že ji čistíme, přesuneme starou databázi a zamítneme LDAPv2 protokol.

```
$ sudo apt update
$ sudo apt install slapd ldap-utils -y
```

Po provedení instalace je potřeba přidat uživatele. To se provede vytvořením textového souboru se všemi podstatnými daty o uživateli a poté se vloží do LDAP adresáře.

```
$ sudo nano ldap_itsme.ldif
```

```
dn: cn=Admins,ou=Groups,dc=ldap,dc=janjavorek,dc=com
objectClass: posixGroup
gidNumber: 5000

dn: uid=itsme,ou=People,dc=ldap,dc=janjavorek,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: itsme
sn: Javorek
givenName: Jan
cn: Jan Javorek
displayName: Itsme
uidNumber: 10000
gidNumber: 5000
```

```
userPassword: itsme
gecos: Jan Javorek
loginShell: /bin/bash
homeDirectory: /ldap_data/itsme
```

Přidání uživatele do LDAP adresáře.

```
$ sudo ldapadd -x -W -D "cn=admin,dc=ldap,dc=janjavorek,dc=com" -f ldap_itsme.
    ldif
```

Teď bychom měli být schopni uživatele v LDAP adresáři vyhledat pomocí následujícího příkazu.

```
$ sudo ldapsearch -x -LLL -b dc=ldap,dc=janjavorek,dc=com 'uid=itsme' cn
    gidNumber
```

Zbývá nastavit oba servery tak, aby se druhý server replikoval oproti tomu prvnímu.

Nastavení Master serveru spočívá ve vytvoření uživatele speciálně pro replikaci, který má přístup ke všem LDAP objektům. To je pro zpřehlednění a také k lepší bezpečnosti. Právě proto není vhodné využívat *root* uživatele.

Vytvoření textového souboru s daty uživatele.

```
dn: cn=rpuser,dc=ldap,dc=janjavorek,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: rpuser
description: Replication User
userPassword: admin
```

Přidání uživatele do LDAP adresáře.

```
$ sudo ldapadd -x -W -D "cn=admin,dc=ldap,dc=janjavorek,dc=com" -f rpuser.ldif
```

Dále je potřeba zapnout modul potřebný pro replikaci. Vytvoříme soubor *syncprov_mod.ldif*.

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib/ldap
olcModuleLoad: syncprov.la
```

Odeslání konfigurace na LDAP server.

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov_mod.ldif
```

Teď potřebujeme zapnout modul pro všechny adresáře. Vytvoříme soubor *syncprov.ldif* a opět odešleme konfiguraci na server.

```
dn: olcOverlay=syncprov,olcDatabase={1}mdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpSessionLog: 100
```

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov.ldif
```

V poslední řadě musíme dát uživateli *rpuser* práva ke čtení celého LDAP serveru. Vytvoříme soubor *syncaccess.ldif* a odešleme konfiguraci.

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to attrs=userPassword by dn="cn=rpuser,dc=ldap,dc=janjavorek,dc=
    com" read by anonymous auth by self write by * none
olcAccess: {1}to attrs=shadowLastChange by self write by * read
olcAccess: {2}to * by * read
```

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f syncaccess.ldif
```

Je třeba se také ujistit, že v souboru */etc/ldap/ldap.conf* je nastavená správně doména.

```
...

BASE dc=ldap,dc=janjavorek,dc=com
URI ldap://localhost

...
```

Nastavení Slave serveru pro replikaci se provede vytvořením jednoho konfiguračního souboru *rp.ldif* se všemi potřebnými daty jako LDAP server, URI, LDAP uživatel, heslo a další. Konfiguraci odešleme na server.

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
provider=ldap://ldap.janjavorek.com/
bindmethod=simple

binddn="cn=rpuser,dc=ldap,dc=janjavorek,dc=com"

credentials=root1234
searchbase="dc=ldap,dc=janjavorek,dc=com"
scope=sub
tls_reqcert=never
timeout=1
schemachecking=on
type=refreshAndPersist
retry="30 5 300 3"
interval=00:00:05:00
```

Nyní restartujeme oba LDAP servery. Konfigurace je dokončena. Poslední částí je tvorba webové aplikace pro reprezentaci obou poskytovatelů služeb a úprava webového prostředí poskytovatele identit.

3.7 Webová aplikace pro SP

Webové aplikace pro SP jsou implementovány s využitím PHP frameworku Nette. V rámci aplikací je vytvořeno jednoduché přihlášení, po kterém se zobrazí informace o jakého poskytovatele služeb se jedná, přihlašovací jméno a celé jméno uživatele. Součástí je tlačítko pro odhlášení.

Vytvoření aplikace se provede stažením frameworku přes nástroj Composer, který spravuje veškeré závislosti v PHP. Projekt pro webovou aplikaci potom vytvoříme pomocí následujícího příkazu. Stažení se provede do složky projektu *nette-sp1*.

```
$ composer create-project nette/web-project nette-sp1
```

Nette je založený na struktuře MVC, tzn. Model-View-Controller. *Model* řeší veškerou logiku aplikace, *View* zobrazuje výstup uživateli a *Controller* je prostředník, který komunikuje a propojuje *View* a *Model*. V případě ukázkové aplikace poskytovatele služeb bylo potřeba vytvořit *View* a *Controller*, veškerá logika přihlašování uživatelů je řešena v simpleSAMLphp. V rámci *Controlleru* se provedlo napojení na simpleSAMLphp, vytvořily se signaly pro přihašovaní, odhlašování a zobrazení požadovaných informací po přihlašení.

Ukázka souboru *Controlleru HomepagePresenter.php*.

```
<?php

declare(strict_types=1);

namespace App\Presenters;
//Napojení na simpleSAMLphp
require_once('../../../../simplesamlphp/lib/_autoload.php');

final class HomepagePresenter extends BasePresenter
{

   public $as;

   function __construct() {
      $this->as = new \SimpleSAML\Auth\Simple('sp1');
   }

   //Signál pro odhlášení
   function handleLogout(){
      if($this->as->isAuthenticated()){
```

```
            $this->as->logout('https://sp1.janjavorek.com/sp_nette/www/');
        }
    }

    //Signál pro přihlášení
    function handleLogin(){
        if(!$this->as->isAuthenticated()){
            $this->as->login([
            'idp' => 'https://sso.janjavorek.com/simplesaml/saml2/idp/metadata.php'
            ]);
        }
        $this->redirect('Homepage:');
    }

    //Zobrazení údajů podle oid pokud je uživatel přihlášený
    public function renderDefault(): void {
        $this->template->login = $this->as->isAuthenticated();
        if($this->as->isAuthenticated()){
            $attributes = $this->as->getAttributes();
            $this->template->userId = $attributes['urn:oid
                :0.9.2342.19200300.100.1.1'][0];
            $this->template->name = $attributes['urn:oid:2.5.4.3'][0];
        }
    }

}
```

Nette poskytuje šablonovací systém pro PHP *latte*, který umožňuje rychlejší a bezpečnější zobrazování dat. Ukázka souboru *View Homepage/default.latte*.

```
{block content}
    <div class="wrap">
        <h3 class="sptitle">Service provider 1</h3>
        {if !$login}
            <a n:href="login!">
                <button class="btn w-100">Přihlásit se do systému</button>
            </a>
        {else}
            <h4 class="splogged">Přihlášení proběhlo úspěšně</h4>
```

```
        <p>Jste přihlášen jako: {$userId}<br>
        Celé jméno: {$name}</p>
        <a n:href="logout!">
           <button class="btn w-100">Odhlásit se</button>
        </a>
      {/if}
   </div>
{/block}
```

Aplikace pro oba poskytovatele služeb jsou velmi jednoduché, avšak jsou připravené pro to, aby mohly být rozšířeny požadovaným specifickým směrem. Například pro internetový obchod nebo jako jiné internetové služby.

3.8 Webová aplikace pro IdP

Webová aplikace pro IdP oproti těm pro SP se implementuje trochu jiným způsobem. Tím, že funkčnost už existuje v rámci knihovny simpleSAMLphp, je přístup tvorby odlišný. Vytváří se pouze vzhled neboli téma této aplikace, které poté systém používá. První věcí je vytvoření simpleSAMLphp modulu */modules/idpmodule* a do něj umístění tématu *idpmodule/themes/idptheme*. Zatím se jedná pouze o vytváření adresářů. Dále je potřeba nastavit simpleSAMLphp, aby nové téma využíval, to se provede v *config.php*

```
...

'theme.use'   => 'idpmodule:idptheme',

...
```

V posledním kroku zkopírujeme soubor *header.php* ze základní šablony a upravíme v něm cestu k CSS souboru.

```
$ cp templates/includes/header.php modules/idpmodule/themes/idptheme/default/
    includes/
```

Téma webové aplikace nyní upravíme pro potřeby poskytovatele identity.

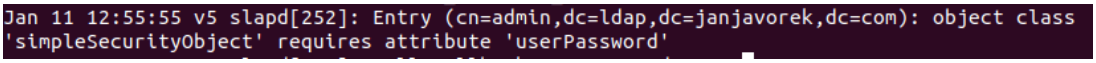
3.9 Problémy při konfiguraci

Při konfiguraci celého projektu nastalo několik problémů. Jeden z nich byl v části nastavování simpleSAMLphp. Konkrétně šlo o část s nalezením správných metadat pro poskytovatele služeb i identity. Metadata zajišťují správný výběr modulů, které se budou používat, správné přesměrování na danou službu, poskytovatele identity, odhlašování a další. Proto je potřeba dávat pozor při jejich nastavování.

Další problém nastal při konfiguraci OpenLDAP replikace. Vedlejší Slave server kopíruje veškeré nastavení a změny od Hlavního Master serveru. To dělá skrze speciálního uživatele, toho je potřeba vytvořit a nastavit mu správná přístupová práva, aby se replikace mohla provádět.

Problém, který přetrvával během celé konfigurace byla práce s LXD/LXC kontejnery. Občas se stávalo, že nástroje v nich běžící padaly, ať už kvůli verzím nebo nastavením.

Přes tyto problémy, které se podařilo úspěšně vyřešit, je projekt funkční a připravený k nasazení do praxe.



Obrázek 10: Záznam chyby při konfiguraci Master-Slave replikace LDAP serverů

4 Ověření funkčnosti systému

4.1 Hodnocení výkonu systému

Tato část se zaměřuje na hodnocení výkonnosti systému a vytížení systémových prostředků. K měření byl použitý nástroj *ctop* a informace přímo z LXC kontejnerů. V tabulce č. 3 jsou hodnoty naměřené při činnosti systému bez jakéhokoli zatížení a tabulka č. 4 zaznamenává hodnoty v průběhu zatížení.

Tabulka 3: Naměřené hodnoty bez zatížení systému

Doména	Využití Paměti [MB]	Odeslané pakety [kB]	Přijaté pakety [kB]
sso.janjavorek.com	154.62/256	3 111	632.03
sp1.janjavorek.com	152/256	17.2	636
sp2.janjavorek.com	211/256	18	639
ldap.janjavorek.com	91/256	25	648
ldap2.janjavorek.com	102/256	32	649

Tabulka 4: Naměřené hodnoty v průběhu zatížení systému

Doména	Využití Paměti [MB]	Odeslané pakety [kB/s]	Přijaté pakety [kB/s]
sso.janjavorek.com	154.62/256	5 980	14 200
sp1.janjavorek.com	152/256	4 330	680.90
sp2.janjavorek.com	211/256	17	639
ldap.janjavorek.com	91/256	25	648
ldap2.janjavorek.com	102/256	32	649

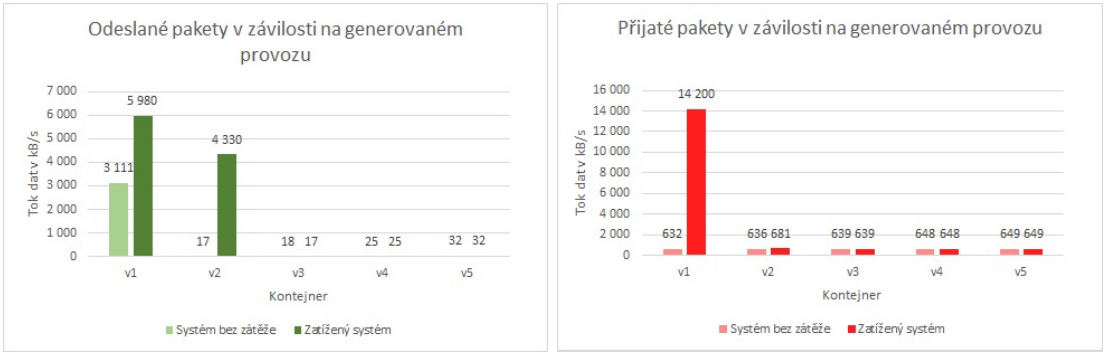
V tabulce č. 5 jsou hodnoty vytížení procesoru v rámci kontejneru *v1* určeného pro poskytovatele identity s doménou *sso.janjavorek.com*.

Testování probíhalo na zařízení s procesorem Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz se čtyřmi jádry. Každý kontejner při své činnosti tedy využíval všechny 4 jádra procesoru. Díky virtualizaci pomocí kontejnerů jsou jednotlivé servery od sebe oddělené a lze měřit výkon procesoru a dalších systémových zdrojů pouze u určitého kontejneru. Každý kontejner má přiděleno 256 MB paměti. Všechny potom běží na systému Ubuntu 18.04. Test spočíval v generování požadavků na přihlášení do systému. Generování požadavků zajišťoval jednoduchý skript vytvořený v *Bashi*. Tyto požadavky se generovaly po dobu 60s a byly odesílané skrze kontejner *v2* tzn. poskytovatele služeb *sp1.janjavorek.com*. Naměřené hodnoty představují nejvyšší dosažené maximu v průběhu této měřené doby.

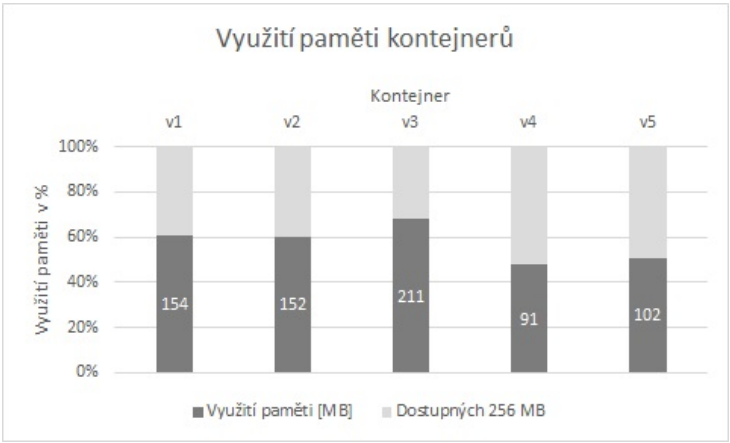
Tabulka 5: Vytížení procesoru v závislosti na zatížení systému pro sso.janjavorek.com

CPU jádro	Bez zatížení		Při zatížení	
	CPU[s]	CPU[%]	CPU[s]	CPU[%]
1	12,4	20,6	12,7	21,2
2	9,2	15,3	9,4	15,6
3	11,5	19,1	11,8	19,6
4	6,9	11,5	7,2	12
Celkově	38,8	64,6	40,17	67

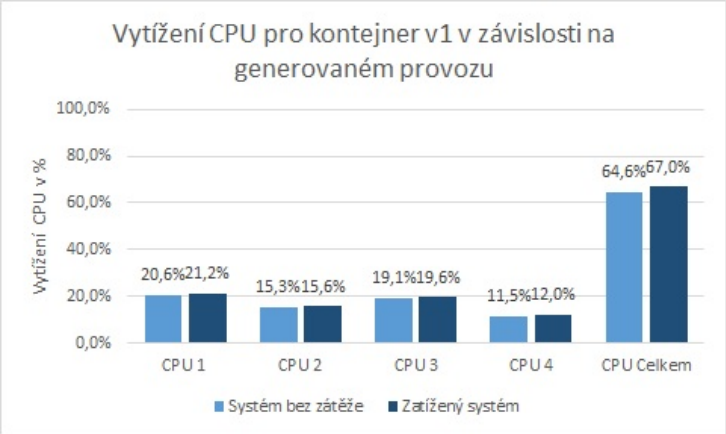
Z naměřených hodnot je patrné, že celková zátěž na procesor v průběhu měření se zvýšila pouze o 2,4 %. Hodnoty jsou zanesené do grafu na obrázku č. 13. Další měřenou částí byly odesílané a přijaté data. K největšímu nárůstu přijatých paketů došlo u kontejneru v1, což je právě IdP, který generované požadavky dostává a ověřuje. Nárůst odesílaných dat je u konteneru v1 a v2, kde dochází ke komunikaci mezi IdP a SP, protože jsou požadavky generovány právě na přístup do systému skrze *sp1.janjavorek.com*. U ostatních kontejnerů nedošlo téměř k žádné změně viz. obrázek č. 11 V grafu na obrázku č. 12 je vidět využití paměti každého kontejneru.



Obrázek 11: Grafy přijímaných a odesílaných dat



Obrázek 12: Graf využití paměti



Obrázek 13: Graf vytížení procesoru

4.2 Využití v praxi

SAML je využívaný celosvětově, mezi organizace využívající tento standard patří například v soukromém sektoru Nokia, Rolls Royce, Google, NTT a spoustu dalších. Ve veřejném sektoru to jsou země napříč evropou, USA až po Čínu. Ty využívají systém v bankovnictví, pro správu průkazů totožnosti a dalších veřejných služeb v rámci státu. SSO autentizační systémy se také využívají v oblasti vzdělávání například pro přístup studentů do školního systému, laboratoří nebo do školních knihoven.

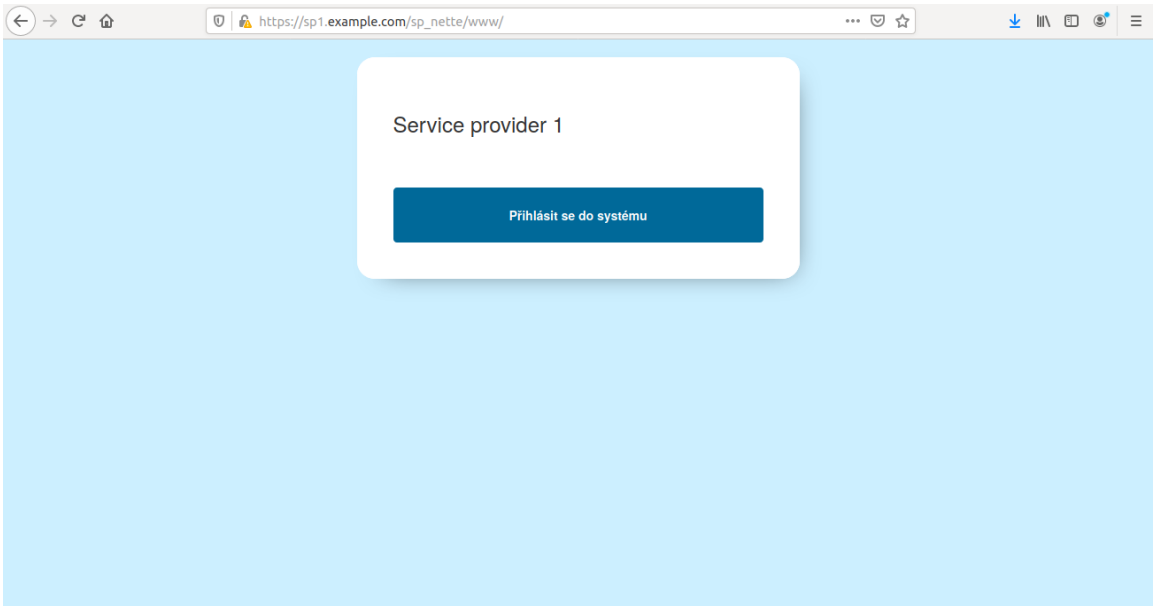
4.3 Možnosti rozšiřování systému

Navržený systém je možné rozšiřovat přidáním více poskytovatelů služeb nebo změnou struktury LDAP serverů. SP je možné přidávat jednoduše po provedení výměny metadat a zařazení SP do konfiguračního souboru na straně IdP. S přidáváním více poskytovatelů služeb roste bezpečnostní riziko, je proto potřeba ověřit tohoto SP, jestli je možné ho považovat za důvěryhodného.

LDAP servery a jejich adresářová struktura se dá měnit na základě potřeb organizace. Je možné například rozdělit tento server dle přístupových práv pro různé skupiny uživatelů. Taky lze toto rozdělení provést způsobem, že skupiny budou odděleny na základě serverů. Jiným možným přístupem může být to, že pokud společnost má více poboček LDAP serverů může být více na různých místech.

4.4 Ukázka funkce SSO systému

Uživatel přijde na webové stránky služby číslo 1 a chce se přihlásit do systému. Po kliknutí na tlačítko přihlásit se do systému je přesměrovaný na poskytovatele identit, v tomto případě *sso.janjavorek.com*.



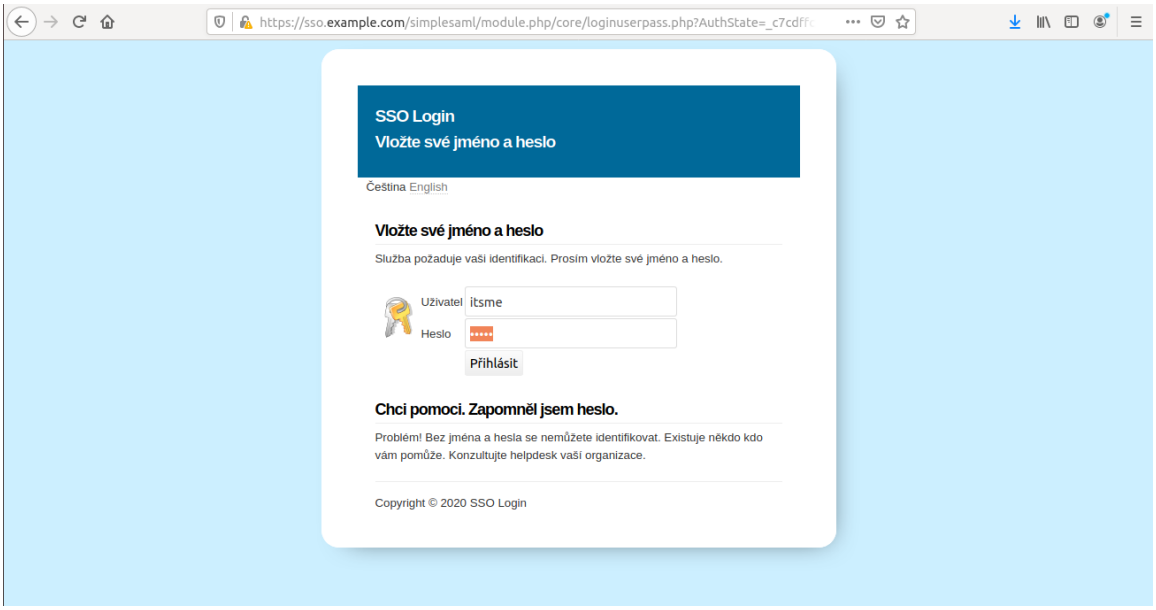
Obrázek 14: Uvítací obrazovka pro službu č.1

Uživatel zadá údaje svého účtu nutné k přihlášení a klikne na tlačítko přihlásit. Zadané údaje se ověří vůči jednomu ze dvou LDAP serverů a přihlásí uživatele. Následně je přesměrován na stránky služby č. 1, ze které zadal požadavek na přihlášení. Viz obrázek č.15.

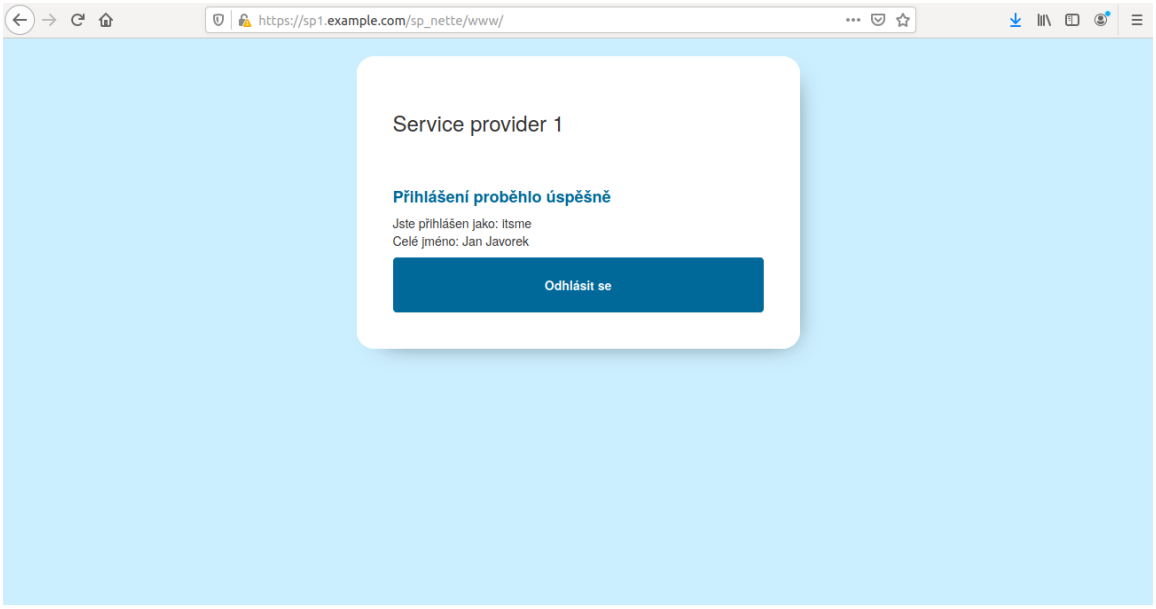
Uživatel je nyní přihlášený v rámci všech služeb napojených na tohoto poskytovatele identity. Konkrétně jak ke službě č.1, tak ke službě č.2. Viz obrázek č.16.

Při přechodu na službu č.2 a kliknutí na tlačítko přihlášení do systému je uživatel přesměrován do systému bez nutnosti znovu zadávat přihlašovací údaje. Viz obrázek č.17 a č.18.

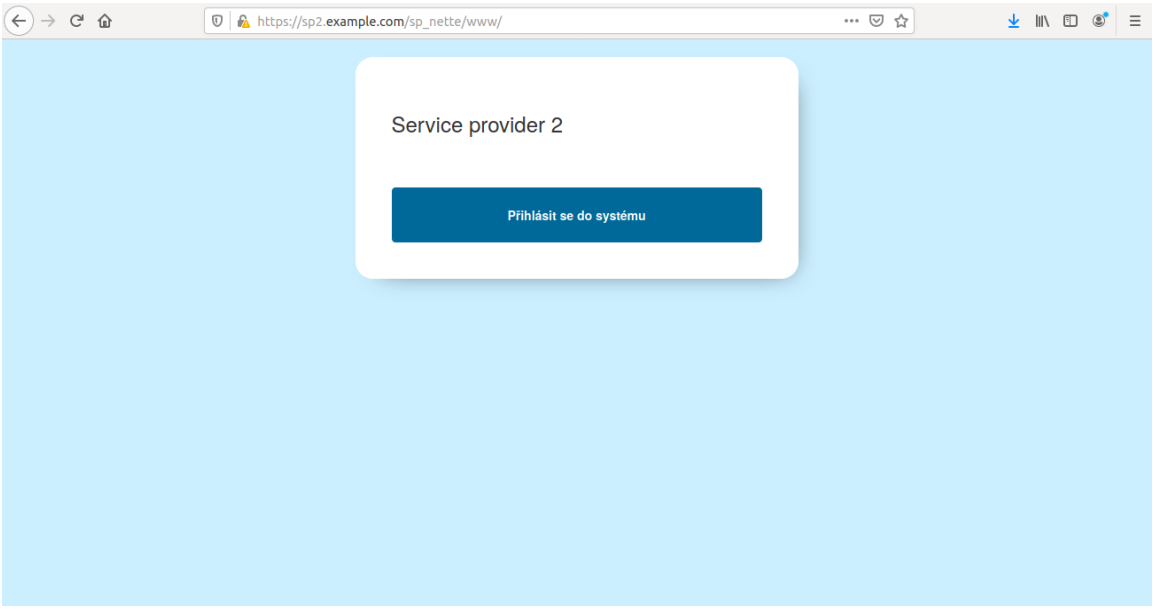
Pokud uživatel klikne na tlačítko odhlásit se, je odhlášen ze všech ostatních služeb, tzn. č.1 i č.2 v rámci daného poskytovatele identity.



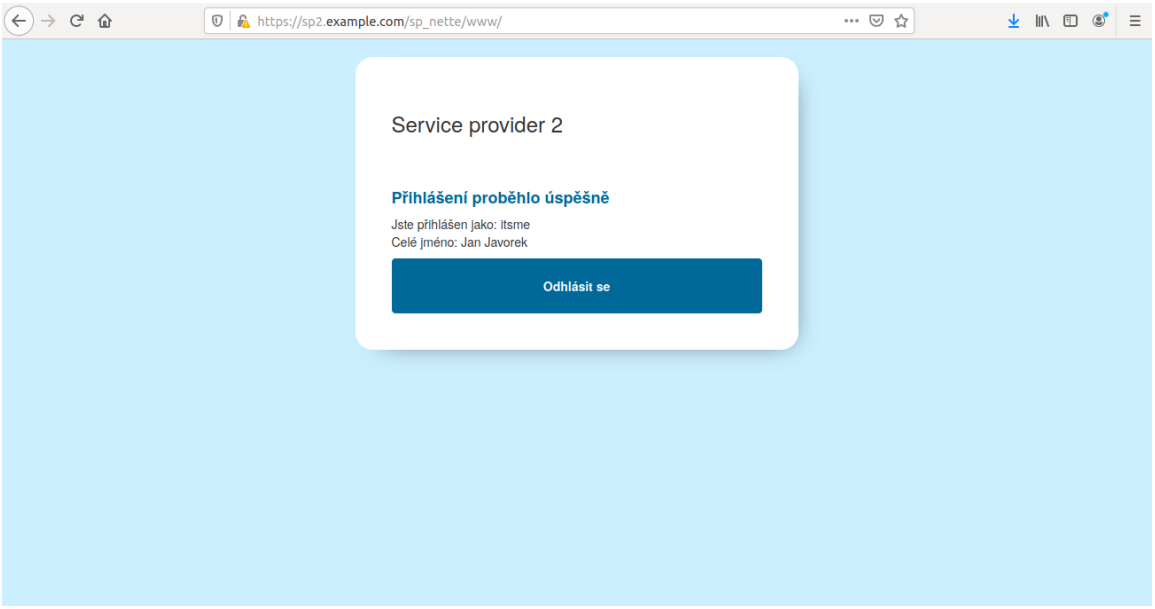
Obrázek 15: Přihlášení do systému



Obrázek 16: Přesměrování na službu č.1 s přístupem do systému



Obrázek 17: Přechod na službu č.2



Obrázek 18: Přístup do systému v rámci služby č.2

5 Závěr

V diplomové práci jsem se věnoval návrhu, vyhotovení a ověření funkčnosti autentizačního systému jednotného přihlašování postaveném na standardu SAML. Součástí práce je popis SSO, standardu SAML a autentizačních metod, dále postup konfigurace a hodnocení výkonu systému.

Velkou výhodou systému jednotného přihlašování je, že usnadňuje uživatelům přístup, protože přihlášení do všech intenetových služeb spojených s daným poskytovatelem identity je pouze jedním heslem. Na druhou stranou to sebou nese riziko, že pokud toto heslo získá útočník dostane se ke všem informacím uživatele.

Systém je vytvořený s využitím virtualizace za pomocí LXC kontejenerů. Kontejnery reprezentují jednotlivé servery v rámci SSO systému, *v1* představuje poskytovatele identity, který zajišťuje autentizaci uživatelů na základě požadavku na přihlášení z příslušné internetové služby reprezentované kontejnery *v2* a *v3*. Po úspěšné autentizaci je uživatel přesměrovaný zpět na službu, ze které požadavek odeslal a má přístup k datům dané služby. Pokud přejde na službu jinou, zahrnutou v tomto systému je mu poskytnut přístup automaticky bez nutnosti se znovu přihlašovat. Autentizace uživatele prováděná poskytovatelem identity je na základě údajů o uživatelích, uložených v adresářové struktuře LDAP serverů, které představují kontejnery *v4* a *v5*. LDAP servery jsou navrženy tak, že v případě výpadku jednoho z nich, je přihlašování do systému stále možné skrze druhý LDAP server. Je to díky nastavení, kdy je adresářová struktura včetně všech změn jednoho serveru duplikována na druhý.

Práce popisuje veškeré použité nástroje potřebné k vyhotovení systému, představení standardu SAML, výhody a rizika využívání SSO systému a různé další metody používané k autentizaci. K implementaci celého systému byla využita, kromě výše zmíněných LXC kontejnerů, také knihovna simpleSAMLphp, která řeší autentizaci právě skrze standard SAML. LDAP servery jsou navrženy za pomoci nástroje OpenLDAP. Webové aplikace jednotlivých poskytovatelů služeb poté využívají Nette framework, který je založený na PHP a zjednodušuje jejich tvorbu.

Dále se práce věnuje podrobnému postupu konfigurace celého systému od nastavení kontejnerů, jednotlivých poskytovatelů služeb a identity, struktury LDAP serverů až po návrh aplikací internetových služeb. Ukázku funkce hotového autentizačního systému je možné vidět v kapitole č. 4.2. V průběhu konfigurace nastalo několik problémů. V první řadě se jednalo o práci s metadaty u nastavování knihovny simpleSAMLphp. Metadata mají velmi důležitou roli u poskytovatelů služeb a identity proto je potřeba dbát na jejich správnost. Další problémy se týkaly práce s LXC kontejnery a konfigurace LDAP serverů tak, aby docházelo k duplikaci adresářové struktury a všech na nich provedených změn.

Systém je možné rozšiřovat přidáním dalších internetových služeb, které budou využívat autentizaci pomocí jednoho poskytovatele. Přidání nových služeb do systému je jednoduše nastavitelné. Další směr rozšiřování systému je změnou struktury LDAP serverů s daty uživatelů. Je možné přidat další takové servery pro autentizaci pouze určité skupiny uživatelů se specifickými přístupovými právy nebo strukturu na základě přístupových práv upravit.

SSO systém byl testovaný na využívání systémových zdrojů. Testování spočívalo v generování požadavků na přihlášení po dobu 60s. Měřilo se využití paměti, výkon procesoru a datový tok. Jednotlivé parametry pak byl měřený v rámci kontejnerů, tzn. využití zdrojů pouze daného kontejneru nikoli celého stroje. Nástroje použité k měření byly ctop a informace přímo z LXC kontejnerů. Měřením bylo zjištěno, že při generování požadavků došlo u kontejneru *v1* resp. poskytovatele identity ke zvýšení výkonu procesoru o 2,4% , odesílání dat se v rámci něj zvýšilo o 2,869 MB/s a zvýšení přijatých dat o 13,568 MB/s. Využití paměti se nezměnilo.

SSO autentizační systémy jsou velmi rozšířené v rámci soukromých i veřejných organizací. Řešení navržené v této práci je funkční a připravené na praktické nasazení, například v rámci internetových obchodů nebo jako systém pro různé společnosti. Systém je uzpůsobený tak, aby šel použít jako šablona pro finální konkrétní aplikaci. Úpravou webových aplikací pro dané služby a případnou změnou struktury LDAP serveru k potřebám organizace, lze vyhotovené řešení plně využít.

Odkazy

[1] Security Assertion Markup Language (SAML) V2.0 Technical Overview, Získané 20. dubna, z `http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html`

[2] stale SimpleSAMLphp Documentation, Získané 20. dubna, z `https://simplesamlphp.org/docs/stable/`

[3] Nette framework dokumentace, Získané 20. dubna, z `https://doc.nette.org/en/3.0/`

[4] php Documentation, Získané 20. dubna, z `https://www.php.net/docs.php`

[5] Lightweight Directory Access Protocol, Získané 20. dubna, z `https://ldap.com`

[6] Learn how to Install LXD / LXC Containers in Ubuntu. *LinuxTechi*, Získané 20. dubna, z `https://www.linuxtechi.com/install-lxd-lxc-containers-from-scratch/`

[7] How to configure OpenLDAP Master-Slave Replication. *ItzGeek*, Získané 20. dubna, z `https://www.itzgeek.com/how-tos/linux/configure-openldap-master-slave-replication.html`

[8] How To Install and Configure SimpleSAMLphp for SAML Authentication on Ubuntu 16.04 *DigitalOcean*, Získané 20. dubna, z `https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-simplesamlphp-for-saml-authentication-on-ubuntu-16-04`

[9] How To Install and Configure SimpleSAMLphp for SAML Authentication on Ubuntu 16.04 *DigitalOcean*, Získané 20. dubna, z `https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-simplesamlphp-for-saml-authentication-on-ubuntu-16-04`

[10] SimpleSAMLphp Documentation, LDAP Module *SimpleSAMLphp*, Získané 20. dubna, z `https://simplesamlphp.org/docs/stable/ldap:ldap`

[11] ZIFČÁK, Jiří. *Jednotný autentizační systém s podporou Kerberosu*[online]. Ostrava, 2015 [cit. 2020-05-14]. Dostupné z: `http://hdl.handle.net/10084/108588`. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.

[12] TAKÁCS, Endre. *Systémy jednotného přihlášení*[online]. Brno, 2011 [cit. 2020-05-14]. Dostupné z: `http://dspace.vutbr.cz/bitstream/handle/11012/71551/final-thesis.pdf`. Bakalářská práce. Vysoké učení technické v Brně.